

# **SPRINT -1**

## **GAS LEAKAGE MONITORING AND ALERTING SYSTEM**

Team ID	PNT2022TMID53713
Project Name	Gas Leakage Monitoring and Alerting System for Industries

### **SIMULATION CREATION USING WOKWI:**

#### **CODE:**

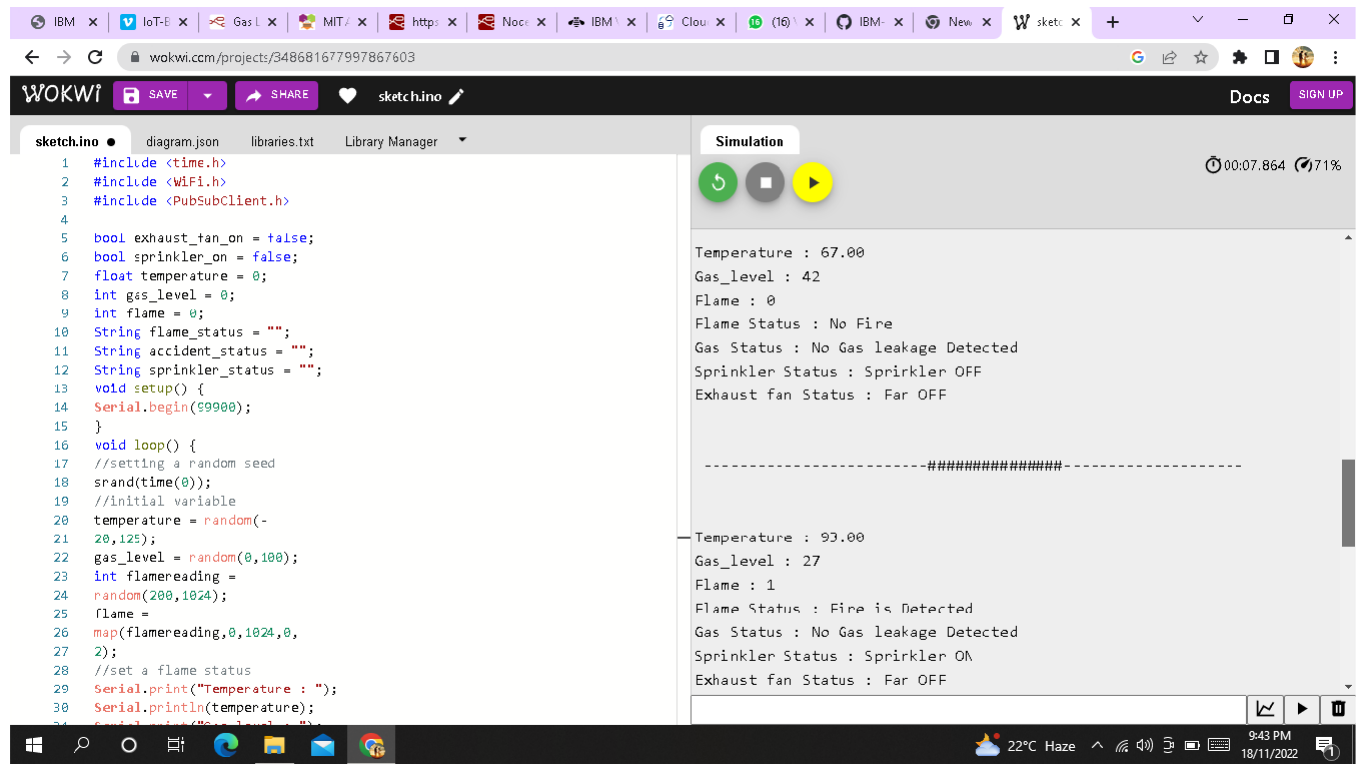
```
#include <time.h> #include
<WiFi.h> #include
<PubSubClient.h>
bool exhaust_fan_on = false;
bool sprinkler_on = false;
float temperature = 0;
int gas_level = 0;
int flame = 0;
String flame_status = "";
String accident_status = "";
String sprinkler_status = "";
void setup() {
  Serial.begin(99900);
}
void loop() {
  //setting a random seed
  srand(time(0));
  //initial variable
  temperature = random(-
  20,125);
  gas_level = random(0,1000);
  int flamereading =
  random(200,1024);
  flame =
  map(flamereading,0,1024,0,
  2);
  //set a flame status
  Serial.print("Temperature : ");
  Serial.println(temperature);
  Serial.print("Gas_level : ");
  Serial.println(gas_level);
  Serial.print("Flame : ");
  Serial.println(flame);
  switch (flame) {
  case 0:
```

```

flame_status = "No Fire"; Serial.println("Flame
Status : "+flame_status);break;
case 1:
flame_status = "Fire is Detected";
Serial.println("Flame Status : "+flame_status);
break;
}
//Gas Detection
if(gas_level > 100){
Serial.println("Gas Status : Gas leakage Detected");
}
else{
exhaust_fan_on = false;
Serial.println("Gas Status : No Gas leakage Detected");
}
//send the sprinkler status
if(flame){ sprinkler_status
= "Sprinkler ON";
Serial.println("Sprinkler Status : "+sprinkler_status);
}
else{
sprinkler_status = "Sprinkler OFF"; Serial.println("Sprinkler
Status : "+sprinkler_status);
}
//toggle the fan according to gas
if(gas_level > 100){
exhaust_fan_on = true;
Serial.println("Exhaust fan Status : Fan ON");
}
else{
exhaust_fan_on = false; Serial.println("Exhaust
fan Status : Fan OFF");
}
Serial.println("");
Serial.println("");
Serial.println(" -----#####----- ");
Serial.println("");
Serial.println("");
delay(1000);
}

```

## SIMULATION OUTPUT:



The screenshot shows the Wokwi IoT simulator interface. On the left, a sketch.ino file is open, displaying C++ code for an IoT device. The code includes libraries for time, WiFi, and PubSubClient. It defines variables for exhaust fan status, sprinkler status, temperature, gas level, flame, and flame status. The setup function initializes the serial port at 999000. The loop function sets a random seed, initializes variables, and generates random values for temperature, gas level, flame, and flame status. It then prints these values to the serial monitor.

On the right, the Simulation panel shows the output of the sketch. The output displays the following data:

```
Temperature : 67.00
Gas_level : 42
Flame : 0
Flame Status : No Fire
Gas Status : No Gas leakage Detected
Sprinkler Status : Sprinkler OFF
Exhaust fan Status : Far OFF
```

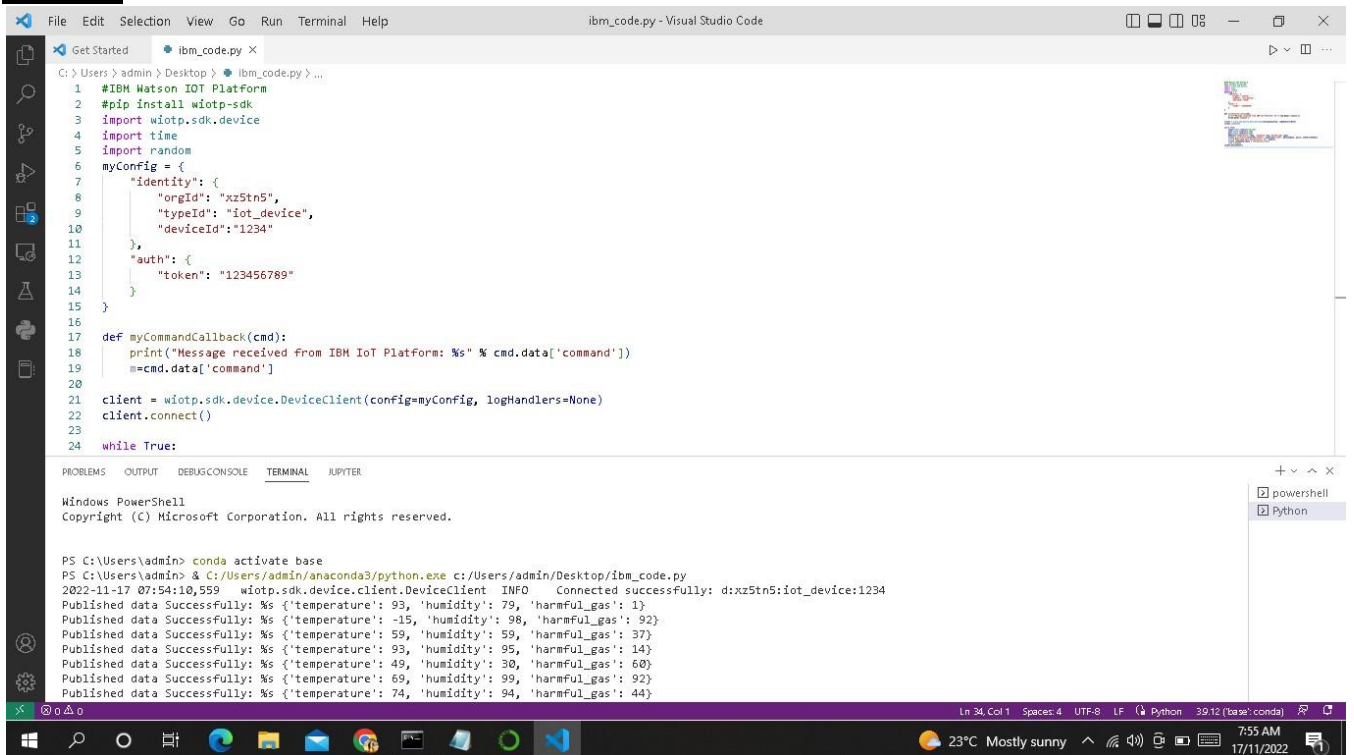
Below this, a separator line of dashes is shown, followed by another set of data:

```
Temperature : 93.00
Gas_level : 27
Flame : 1
Flame Status : Fire is Detected
Gas Status : No Gas leakage Detected
Sprinkler Status : Sprinkler ON
Exhaust fan Status : Far OFF
```

The bottom of the interface shows the Windows taskbar with the system clock at 9:43 PM on 18/11/2022.

## CONNECTING IBM CLOUD USING PYTHON CODE:

### CODE:



The screenshot shows a Visual Studio Code editor window with a Python script named `ibm_code.py`. The script is designed to connect to the IBM IoT Platform and publish data. It includes the following code:

```
1 #IBM Watson IOT Platform
2 #pip install wiotp-sdk
3 import wiotp.sdk.device
4 import time
5 import random
6 myConfig = {
7     "identity": {
8         "orgId": "xz5tn5",
9         "typeId": "iot_device",
10        "deviceId": "1234"
11    },
12    "auth": {
13        "token": "123456789"
14    }
15 }
16
17 def myCommandCallback(cmd):
18     print("Message received from IBM IoT Platform: %s" % cmd.data['command'])
19     #cmd.data['command']
20
21 client = wiotp.sdk.device.DeviceClient(config=myConfig, logHandlers=None)
22 client.connect()
23
24 while True:
```

The bottom of the editor shows the terminal output, which displays the following messages:

```
PS C:\Users\admin> conda activate base
PS C:\Users\admin> & C:\Users\admin\anaconda3\python.exe c:/Users/admin/Desktop/ibm_code.py
2022-11-17 07:54:10.559 wiotp.sdk.device.client.DeviceClient INFO Connected successfully: d:xz5tn5:iot_device:1234
Published data Successfully: %s {'temperature': 93, 'humidity': 79, 'harmful_gas': 1}
Published data Successfully: %s {'temperature': -15, 'humidity': 98, 'harmful_gas': 92}
Published data Successfully: %s {'temperature': 59, 'humidity': 59, 'harmful_gas': 37}
Published data Successfully: %s {'temperature': 93, 'humidity': 95, 'harmful_gas': 14}
Published data Successfully: %s {'temperature': 49, 'humidity': 30, 'harmful_gas': 60}
Published data Successfully: %s {'temperature': 69, 'humidity': 99, 'harmful_gas': 92}
Published data Successfully: %s {'temperature': 74, 'humidity': 94, 'harmful_gas': 44}
```

The bottom of the image shows the Windows taskbar with the system clock at 7:55 AM on 17/11/2022.

## OUTPUT IN IBM CLOUD:

The screenshot displays the IBM Cloud IoT Dashboard interface. At the top, there are navigation tabs: 'Browse', 'Action', 'Device Types', and 'Interfaces'. On the right, there is an 'Add Device' button. The main content area shows details for a specific device with ID '1234', which is 'Connected' and of type 'iot\_device'. The device was last updated on 'Nov 5, 2022 9:04 PM'. Below this, there are tabs for 'Identity', 'Device Information', 'Recent Events', 'State', and 'Logs'. The 'Recent Events' tab is selected, showing a list of events. A message states: 'The recent events listed show the live stream of data that is coming and going from this device.' The events are listed in a table with columns: 'Event', 'Value', 'Format', and 'Last Received'. The table contains five entries, all with the event type 'status' and a 'json' format. The 'Value' column contains truncated JSON strings representing sensor data. The 'Last Received' column indicates that each event was received 'a few seconds ago'. At the bottom left, there is a pagination control showing 'Items per page 50' and '1-2 of 2 items'. At the bottom right, a status box indicates '0 Simulations running'.

1234 Connected iot\_device Device Nov 5, 2022 9:04 PM

Identity Device Information **Recent Events** State Logs

The recent events listed show the live stream of data that is coming and going from this device.

Event	Value	Format	Last Received
status	{"temperature":49,"humidity":30,"harmful_gas"...	json	a few seconds ago
status	{"temperature":93,"humidity":95,"harmful_gas"...	json	a few seconds ago
status	{"temperature":59,"humidity":59,"harmful_gas"...	json	a few seconds ago
status	{"temperature":-15,"humidity":98,"harmful_gas"...	json	a few seconds ago
status	{"temperature":93,"humidity":79,"harmful_gas"...	json	a few seconds ago

Items per page 50 | 1-2 of 2 items

0 Simulations running